

OPTIMAL PUSHBACK TIME WITH EXISTING UNCERTAINTIES AT BUSY AIRPORT

Ryota Mori*

*Electronic Navigation Research Institute

Keywords: TSAT, reinforcement learning, uncertainty

Abstract

Pushback time management of departure aircraft is a promising method to reduce the fuel burn during airport ground operations. Departure aircraft can wait at the gate with engines off instead of waiting in a long queue before the runway. However, this management can potentially delay the take-off time, which in turn reduces the capacity and passenger satisfaction. Therefore, pushback time management strategy should be applied as long as the imposed negative effect is sufficiently small. This paper proposes an optimal pushback time assignment strategy by considering reduction of both fuel burn and delay. The optimal strategy is obtained via Monte Carlo reinforcement learning, and it provides better performance than the manual tuning. In addition, the interpretation of the obtained strategy provides further knowledge about airport's traffic. The proposed method can also potentially provide better rule-based strategy.

1 Introduction

Airport congestions have recently become a critical problem at many airports in the world. The bottleneck of airport operations is usually found on the runway, because the number of take-off and landing aircraft is limited due to the required minimum aircraft separation. As a result, there are long waiting queues of aircraft both on the ground taxiway and in the air, which leads to an increase in both fuel burn and emissions. Arrival aircraft are often a target of airport congestion research because any additional flight time obviously requires extra fuel. Departure aircraft, though, also burn

sufficient amount of fuel during taxiing, too, so departure queue management can further help to reduce fuel burn. Besides, compared to arrival queues, departure queues can be controlled more easily by allocating appropriate pushback time. The allocated pushback time is called TSAT (Target Start Approved Time). Departure aircraft waiting at the gate can stop its engines and therefore save fuel.

Fuel saving has been the main issue for efficient airport operation, and there are several approaches to allocate the appropriate pushback time. Sandberg et al propose N -control strategy, which limits the number of aircraft on the ground, a strategy proven to work at Boston airport already[1]. Smeltink et al propose a TSAT allocation algorithm based on mixed integer programming[2]. Atkin et al propose a TSAT allocation algorithm based on rolling window approach and consider the effect of TSAT allocation to aircraft sequencing problem[3][4]. The objective function is usually set as total taxi-out time, but Ravizza et al discuss the trade-off between taxi-out time and fuel consumption by TSAT setting[5]. However, these works have not considered allocations' potential negative effect, e.g. aircraft delay. If the pushback time is controlled to save taxiing time, allocated pushback time is usually later than the expected pushback time, which means that the take-off time can be delayed due to various uncertainties. In this research, "delay" is defined as the difference of take-off time between the nominal case and the pushback time controlled case. Even if a large margin is set to absorb uncertainties, the expected delay will be close to 0, but not definitely 0. According to most researches mentioned above, the taxiing time is reduced as long as further

delay is not caused, but strictly speaking, this cannot happen. Even if delay is caused, it is usually attributed to a “rare unexpected event”. Although the fuel saving is an obvious advantage for most stakeholders, the definition of “good” airport operation varies with stakeholders. For example, airlines are concerned about both take-off delay and fuel saving. If the take-off time is delayed, the arrival time will most probably also be delayed or more fuel will have to be consumed for speeding-up the aircraft to arrive on time. Therefore, the “real” optimal airport operation should be discussed from various perspectives.

This research is unique in terms of considering important factors for various stakeholders, not just considering fuel saving only. In the previous research[6], it was proven that two factors (fuel saving and take-off delay) had actually trade-off relationship, but the TSAT allocation algorithm used there was only in its preliminary version. This paper improves the TSAT allocation algorithm using easily obtained information via Monte Carlo reinforcement learning strategy.

2 Overview of Airport Simulation Model

2.1 Tokyo International Airport operation

Tokyo International Airport is the busiest airport in Japan with more than 1,000 departures and arrivals every day, and so it has been chosen as the target airport of this research. There are four runways at this airport, and Fig. 1 shows the runway operation under north wind. Due to the layout of the runways, not all runways can be used simultaneously, as runway dependency exists. When arrival aircraft approaches C runway, the departure aircraft from D runway as well as C runway cannot take off. Under this complex situation, first an airport simulation model is developed, and then TSAT allocation algorithm is discussed.

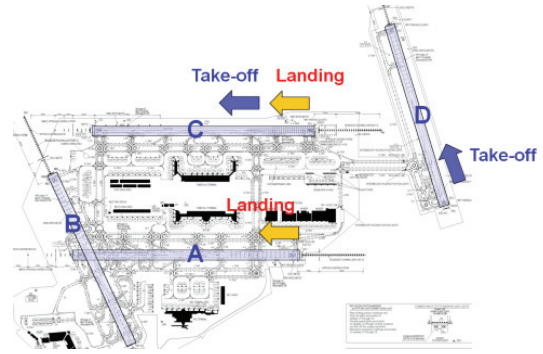


Fig. 1 Current runway operation under north wind.

2.2 Operation flow of departure and arrival aircraft at the airport

In order to model airport operation correctly, it is important to understand the operational flow of departure and arrival aircraft, so here it is explained briefly.

Fig. 2 summarizes the ATC flow of both departure and arrival aircraft. First, the departure aircraft is considered. There are some steps for take-off. First, usually about 5 minutes before the pushback ready ($\Delta t_{clearance}$), the pilot contacts clearance delivery, and gets a departure clearance. Once the aircraft is actually ready for pushback, the pilot contacts ground ATC to get a pushback clearance. When the pushback clearance is obtained, the aircraft starts pushback and contacts ground ATC to get a taxiing clearance. Once the taxiing clearance is obtained, the aircraft goes taxiing to the runway. Then the pilot contacts tower ATC to get a take-off clearance. Only once the take-off clearance is obtained, can the aircraft take off.

There are several events until take-off, so the time is split into several stages ($\Delta t_{pushback}$, $\Delta t_{prepare}$, Δt_{taxi} , and Δt_{wait}), and some variables are defined. The time when the aircraft is ready for pushback is defined as ARDT (Actual Ready Time) and AOBT (Actual Off Block Time) is defined as the pushback start time. When TSAT is not allocated, the aircraft starts pushback just after the aircraft is ready for pushback, i.e. AOBT is usually equal to ARDT.

Next, the take-off time is defined as ATOT (Actual Take-Off Time). The difference between AOBT and ATOT is defined as taxi-out time, AXOT (Actual Taxi-Out Time). When

the runway is congested, the aircraft waits in a departure runway queue. The time spent in a queue can be potentially reduced and defined as Δt_{wait} . When TSAT allocation works well, AOBT is shifted later i.e. $AOBT > ARDT$, and Δt_{wait} is reduced, and in turn the taxi-out time can be reduced. Taxi-out time starts at the pushback start, so TSAT should be allocated prior to pushback start. Here, it is assumed that TSAT is allocated when the pilot contacts a clearance delivery. If TSAT is allocated, the departure aircraft cannot start pushback until TSAT even if the aircraft is ready for pushback.

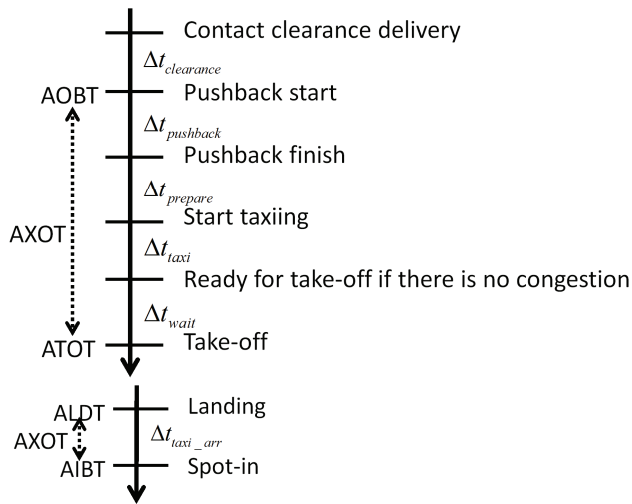


Fig. 2 Flow of departure and arrival aircraft.

Next, arrival aircraft is considered. Since this paper focuses on the airport ground operation only, the operation flow after landing is discussed. Certain time before landing, the estimated landing time (ELDT) is provided to ATC. Once the aircraft lands, the aircraft contacts ground ATC to get a taxiing clearance. The aircraft goes taxiing to the assigned spot, and it gets into the spot as long as there is no aircraft blocking the taxiway or spot. Here, several variables are defined. The time when the aircraft lands is defined as ALDT (Actual Landing Time), and AIBT (Actual In-Block Time) is defined as the time when the aircraft gets into the spot. The time between ALDT and AIBT is defined as AXIT (Actual Taxi-In Time). When TSAT is assigned to departure aircraft, the arrival aircraft sometimes cannot get into the spot due to the departure aircraft. Even if the

taxi-out time is reduced, it is meaningless if the taxi-in time increases. Therefore, both departure and arrival aircraft should be considered in TSAT allocation.

2.3 Overview of airport simulation model

The details of the simulation model are described in Ref. [6], so this paper only briefly presents the essence of the model. This research considers uncertainty effect, so the simulation model includes stochastic process. Several variables ($\Delta t_{clearance}$, $\Delta t_{pushback}$, $\Delta t_{prepare}$, Δt_{taxi} , and Δt_{taxi_arr}) are estimated from actual data, and regression model is obtained with dependent variable. For example, the dependent variable of Δt_{taxi} is the taxiing distance. Then, the residual is modeled by distribution function, such as normal distribution. Δt_{wait} is calculated based on first-come-first-served policy. The take-off and landing separation is also distributed randomly. AOBT and ALDT are usually decided based on the schedule, not totally distributed randomly, so AOBT and ALDT are obtained via actual data and fixed in the simulation as a scenario. Since there are some stochastic parameters, ATOT and AIBT are stochastically distributed.

3 Pushback Time Assigning Algorithm (TSAT)

3.1 Overview of TSAT allocation and its issues

TSAT is assigned to reduce taxiing time of departure aircraft by avoiding waiting in a departure queue. This reduction of waiting time is achieved by shifting the pushback time later, but inappropriate assignment can cause take-off time delay without reducing waiting time. Consider a simple case. If there is only one aircraft at the airport and the pushback time is shifted later, the taxi-out time is not changed and the take-off time is just shifted later. Here, the delay is defined as the difference of take-off time between nominal case (TSAT not allocated case) and TSAT allocated case. Since TSAT just delays the pushback time, not advancing the

pushback time, the delay is usually zero on average at least (TSAT usually does not advance the take-off time on average). Since airlines are concerned with delay as well as fuel saving, TSAT should be allocated to maximize the fuel saving and to minimize the delay. However, there is usually a trade-off between these two factors, so they both should be considered at the same time.

TSAT is usually allocated based on the estimation of runway queueing. If all situations were exactly estimated, maximum fuel saving and zero delay would be achieved. However, due to the existence of uncertainty of various situations, the runway queueing is not completely estimated in advance, which in turn causes delay. To reduce delay, the usual approach is putting some margin to absorb uncertainty. When the assigned take-off time is assumed to be TTOT (Target Take-Off Time), and the estimated taxi-out time is defined as EXOT (Estimated Taxi-Out Time), TSAT is calculated based on the following equation.

$$TSAT = TTOT - EXOT - m \quad (1)$$

where m is margin delay. Note that EXOT does not include any delay margin. TSAT allocation problem can be converted to assigning margin to each aircraft. When $m = \infty$, TSAT does not work, and when $m = 0$, large reduction of taxi-out time is expected (actually the large delay is caused at the same time). If this margin is set appropriately according to the situation, both large reduction of taxi-out time and small delay might be achieved. The details will be given in Sec. 3.3.

To assign TSAT to each aircraft, TTOT and EXOT as well as margin are required according to Eq. (1). EXOT is easily obtained via actual data. In the simulation, the taxi-out time is calculated by the average taxi-out time in each spot and uncertainty, but EXOT can be set the same as the average taxi-out time.

As for TTOT, TTOT is usually decided based on ETOT (Estimated Take-Off Time) and the estimated runway sequencing. Using the data of ETOT of each aircraft, all aircraft are sequenced in each runway in advance and take-off time is allocated to each aircraft. This process is called pre-departure sequencing. The

details will be explained in the next subsection (Sec. 3.2). ETOT is calculated by EOBT (Estimated Off-Block Time) plus EXOT. This flow is summarized in Fig. 3.

All the “estimated” time includes uncertainty, and its accuracy is usually dependent on the timing of data obtained. As previously explained, TSAT is assumed to be allocated when the pilot contacts clearance delivery, so EOBT can be estimated relatively accurately. Here, EOBT is assumed to be equal to ARDT.

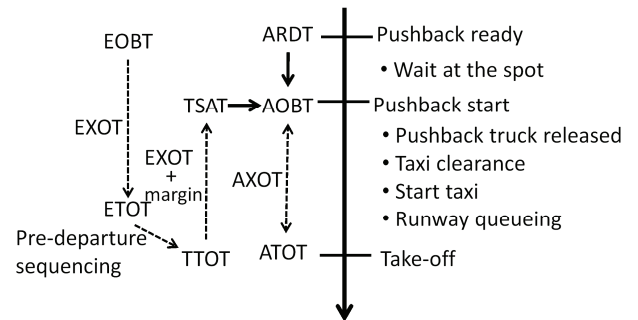


Fig. 3 TSAT allocation flow.

3.2 Pre-departure sequencing

Pre-departure sequencing is important for TSAT allocation, and here TTOT allocation process is explained. It is assumed that each runway has a virtual runway slot for take-off or landing at the constant interval (here 90 s), and a single slot can contain a single aircraft only. As for departure aircraft, when the departure aircraft contacts clearance delivery, the aircraft gets an earliest available runway slot after ETOT. The corresponding time of the obtained slot is set as TTOT. Therefore, TTOT is the same as ETOT earliest, and set later if the runway is congested. However, both C and D runways are affected by landing aircraft on C runway, so landing aircraft should also be considered. As for arrival aircraft, here ELDT is assumed to be available 30 minutes before landing. When ELDT is obtained, the runway slot just after ELDT is assigned to this arrival aircraft. All departure aircraft after this slot is shifted later. Note that TSAT is allocated just after TTOT is allocated, so the slot shift does not affect TTOT of the departure aircraft.

3.3 Margin setting and required information

Once TTOT is determined based on the algorithm explained in Sec 3.2, margin is required to set TSAT. This section provides the margin setting flow. Basically, small margin increases take-off delay and taxiing time reduction is also small, and vice versa. However, better performance (i.e. smaller delay and larger taxiing time reduction) is expected by setting appropriate margin in each situation. The “situation” definition is ambiguous, so here explanatory variable is introduced to help determine the margin.

The variable should contain the information which affects likelihood of take-off delay and reduction of taxiing time. This time, a variable defined as “ n_{ahead} ” is used.

n_{ahead} : The number of aircraft ahead consecutively in a virtual runway slot. ($n_{ahead} \geq 0$)

Large n_{ahead} means that many aircraft are already expected to wait in a departure queue, so the delay is likely to be absorbed by other aircraft ahead. Therefore, large margin is set when n_{ahead} is small, and small margin is set when n_{ahead} is large, so both large taxiing time reduction and small take-of delay are expected to be obtained. Better performance will be expected if additional useful information is used, but here only single information (n_{ahead}) is used.

To be summarized, the margin (m) is set based on the following expression.

$$m = f(n_{ahead}) \quad (0 \leq m \leq \infty) \quad (2)$$

n_{ahead} is an integer, and the optimal margin setting is the same as finding the optimal function f .

3.4 Manual margin tuning

First, f is found based on manual tuning. As written in the previous section, large margin should be set when n_{ahead} is small, and vice versa, so the following function is used.

$$f(n_{ahead}) = \begin{cases} \infty & (0 \leq n_{ahead} < n) \\ x & (n \leq n_{ahead}) \end{cases} \quad (3)$$

This expression means that TSAT is set only when n_{ahead} is n or larger with n and x being the parameters. Based on the manual tuning, n is set to 7, and x is varied between 2 and 6 minutes.

3.5 Simulation results

The simulation environment is explained before showing the simulation results. The simulation needs “scenario” assumption which describes the number of departure and arrival at a certain interval, because the degree of congestion depends on the traffic volume mainly. This time the scenario is made based on the actual operation, and ARDT (pushback ready time of departure aircraft), ALDT (actual landing time), and the spot position of all aircraft are set the same as those in the actual operation. Even if the traffic volume is the same, the distribution of departure and arrival between the time ranges also affects the congestion level, so it is also kept in accordance with the actual operation. This time, data of 15 days are used, and the average taxiing time saving and average take-off delay are considered.

Fig. 4 shows the simulation result. As mentioned in Sec 3.4, x is changed between 2 and 6 minutes. When x is large, large taxi-out time saving is expected but large delay is also caused at the same time, and vice versa. Therefore, there is actually a trade-off between taxi-out time saving and delay. However, this result is just based on manual tuning, so better result might be obtained if the function f is set more appropriately.

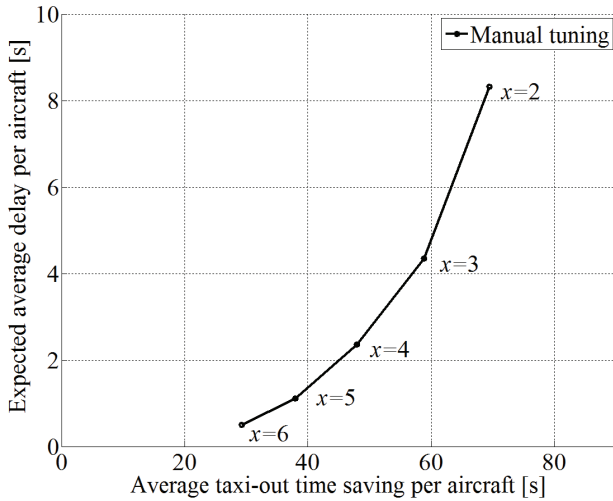


Fig. 4 Relationship between average delay and average taxi-out time saving.

4 Optimal TSAT assignment via reinforcement learning

4.1 Monte Carlo reinforcement learning

To obtain a better strategy of TSAT assignment, i.e. to find a better function of f , Monte Carlo reinforcement learning is applied[7]. Reinforcement learning is a method to obtain the best actions in the environment so as to maximize the objective function. Monte Carlo reinforcement learning is one of the reinforcement learning methods, and here it is briefly explained.

First, an objective function is defined. In this case, the objective function consists of both taxi-out time saving (Δt_{save}) and take-off delay (Δt_{delay}). The objective function (reward) r is set by the following equation.

$$r = \Delta t_{save} - \beta \Delta t_{delay} \quad (4)$$

where β is the weight parameter.

Next, several variables are defined. The state of the environment (this case, n_{ahead}) is defined as s , and the action to be taken (this case, $f(n_{ahead})$) is defined as a . The expected reward when the state is s and the action is a is defined as $Q(s, a)$. $Q(s, a)$ is updated through a simulation. In this case, there are 20 states and 6 discrete actions are assumed as follows.

$$s_0 : n_{ahead} = 0, s_1 : n_{ahead} = 1, \dots, s_{19} : n_{ahead} \geq 19 \quad (5)$$

$$a_0 : f(n_{ahead}) = 0 \quad (6)$$

$$a_1 : f(n_{ahead}) = 2$$

$$a_2 : f(n_{ahead}) = 4$$

$$a_3 : f(n_{ahead}) = 6$$

$$a_4 : f(n_{ahead}) = 8$$

$$a_5 : f(n_{ahead}) = \infty$$

Therefore, there are a total of 120 ($=6 \times 20$) expected rewards, i.e. $Q(s, a)$. Here, the policy (π) is defined as a set of actions in each state, and described by the following equation.

$$\pi = \{\pi(s_0), \pi(s_1), \dots, \pi(s_{19})\} \quad (7)$$

$$\pi(s_i) = a_0 \text{ or } a_1 \text{ or } \dots \text{ or } a_5$$

The flow of reinforcement learning is shown in Fig. 5.

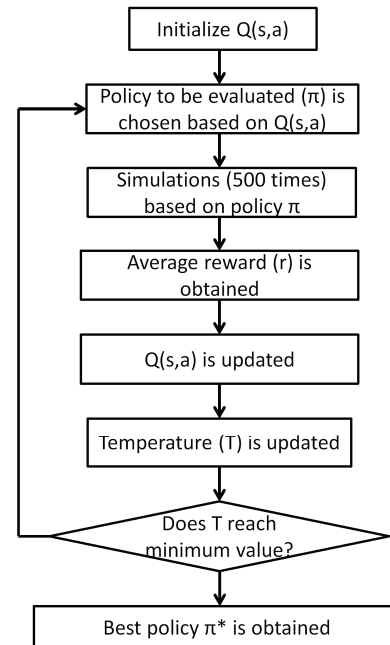


Fig. 5 Flow of Monte Carlo reinforcement learning.

First, $Q(s, a)$ is initialized. The initial value is obtained by 1000 times of simulations with random policies, and all $Q(s, a)$ are set to the average reward. Next, the policy to be evaluated (π) is chosen based on $Q(s, a)$. $Q(s, a)$ is iteratively updated, so a better action in each state has a larger $Q(s, a)$. To reduce

learning time, the action with large $Q(s, a)$ is chosen more often than the one with small $Q(s, a)$. However, to avoid local minimum, the temperature T is introduced. Since each action is more equally chosen with large T , at the beginning of the iteration, large T is set and it is gradually decreased. Finally, the action in each state is chosen with the following probability.

$$P(s_i, a_j) = \frac{\exp(Q(s_i, a_j)/T)}{\sum_k \exp(Q(s_i, a_k)/T)} \quad (8)$$

Based on this probability, the policy to be evaluated is chosen. Next, simulations are conducted 500 times with the same policy. The simulation includes uncertainty effect, so the obtained reward in a single simulation changes even with the same policy. Therefore, simulations are conducted several times, and the average reward is used. Next, $Q(s, a)$ is updated. Assuming that the average reward is r_i in the i th simulation, the expected reward is updated based on the following equation.

$$Q_{i+1}(s, a) = Q_i(s, a) + \alpha(r_i - Q_i(s, a)) \quad (9)$$

α is the parameter of learning rate. Next, T is also updated based on the following equation with the parameter δ .

$$T_{i+1} = \delta T_i \quad (0 < \delta < 1) \quad (10)$$

Finally, after sufficient number of iterations are conducted, the best policy is decided by the following equation.

$$\pi(s) = \arg \max_a (Q(s, a)) \quad (11)$$

4.2 Simulation conditions

Based on the method described in the previous section, it is investigated how much performance improvement is observed. The parameters used in the reinforcement learning are summarized in Table 1. To obtain the trade-off result as shown in Fig. 4, four different learnings are conducted with various β .

Table 1 Parameters in reinforcement learning.

Parameters	Values
β	3, 5, 10, 20
T_0	10.0
T_{\min}	0.05
δ	0.98
α	0.01

4.3 Simulation results

The relationship between taxi-out time saving and delay under various β is shown in Fig. 6. For comparison, the result by manual tuning is also provided. The larger taxi-out time saving and the smaller delay indicates better results. As seen in the figure, the result obtained by reinforcement learning provides better results than that by manual tuning. Especially the case when β is 20, little delay is observed while taxi-out time is saved by about 35 s on average. This means that TSAT can be introduced with little negative impact.

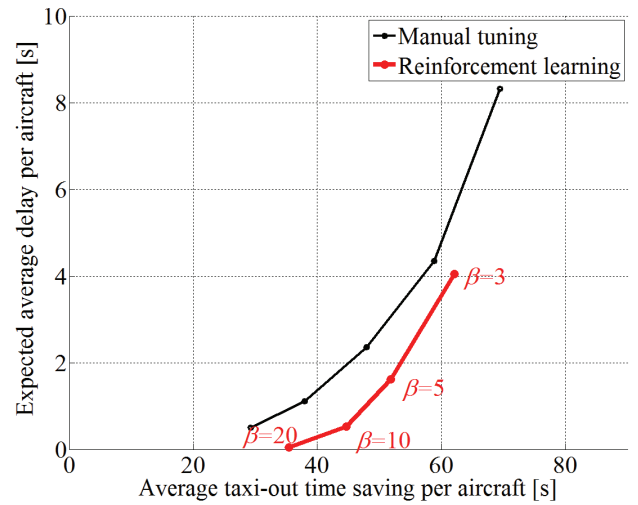


Fig. 6 Relationship between average delay and average taxi-out time saving obtained by reinforcement learning and manual tuning.

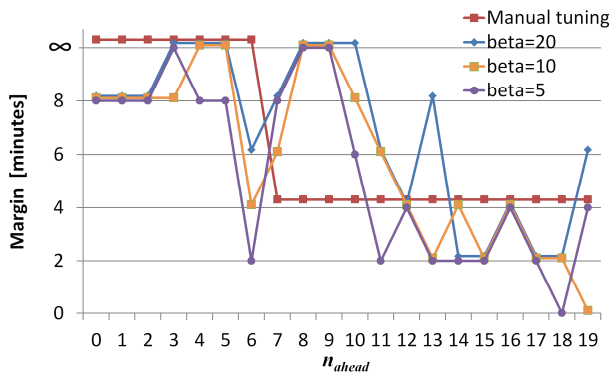


Fig. 7 The obtained policy by reinforcement learning. (Margin is slightly shifted for better understanding)

Next, the obtained policy via reinforcement learning is shown in Fig. 7. Larger margin indicates the smaller taxi-out time saving and smaller delay. Large β relatively weights much on reducing delay, so smaller margins for smaller β are observed. Compared to the case of manual tuning, it is interesting that there is a falling point of margin at $n_{ahead} = 6$ and the margin increases largely for 7 and 8 of n_{ahead} for all cases by reinforcement learning. The possible reason is as follows. Here, “delay” is the average delay per aircraft, so large margin should be set when many aircraft are expected to be queued after the considered aircraft. Fig. 8 shows the frequency of queue length in 20,000 times simulation for 15 days. A longer queue is usually less often observed, so the frequency usually decreases with queue length. However, according to the figure, the frequency of queue length being 10 – 15 is almost the same, which means that these queue lengths are relatively often observed. To avoid the delay increase, the large margin should be set before the queue length gets 10 – 15, i.e. n_{ahead} is about 7 – 9. If the traffic is totally random, longer queues should be observed less often, so this phenomenon is due to the departure and arrival traffic pattern at this airport.

Using the obtained policy, the better performance of TSAT is expected, but at the same time, a new knowledge about the airport traffic is obtained by interpreting the result. The obtained policy is not only the optimal policy,

but also used for the strategy to reduce congestion.

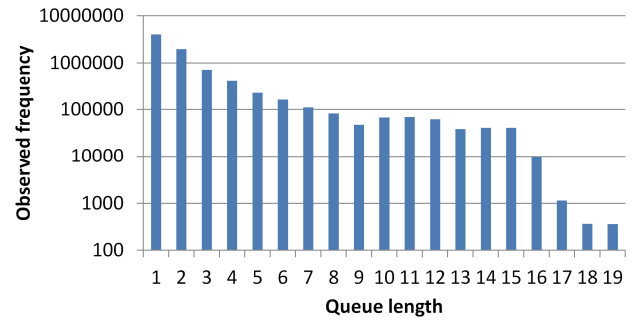


Fig. 8 Observed frequency of queue length. (log scale)

5 Conclusions

To reduce the fuel burn, this paper focused on airport congestion and TSAT allocation algorithm was considered. Since TSAT potentially causes delay as well as reducing taxi-out time, both factors were considered. TSAT allocation algorithm was optimized via reinforcement learning and performance improvement was observed especially for small expected average delay. In addition, the obtained optimal policy via reinforcement learning provided a new knowledge about the airport traffic, which would help reduce the congestion itself. However, the current reinforcement learning uses limited information, so further useful information is to be used for reinforcement learning to improve the performance in the future. Also, the reinforcement learning algorithm itself has still a room for improvement, which will also be a future work.

References

- [1] Sandberg, M., Simaiakis, I., Balakrishnan, H., Reynolds, T. G., and Hansman, R.J., “A Decision Support Tool for the Pushback Rate Control of Airport Departures,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Human-Machine Interaction*, in press.
- [2] Smeltink, J. W., Soomer, M. J., de Waal, P. R., van der Mei, R. D., “An Optimisation Model for Airport Taxi Scheduling,” *Proceedings of the INFORMS Annual Meeting*, 2004.

- [3] Atkin, J. A. D., Maera, G. D., Burke, E. K., Greenwood, J. S., “Addressing the Pushback Time Allocation at Heathrow Airport,” *Transportation Science*, Vol. 47, No. 4, pp. 584-602, 2013.
- [4] Atkin, J. A. D., Burke, E. K., and Greenwood, J. S., “A comparison of two methods for reducing take-off delay at London Heathrow airport,” *Journal of Scheduling*, Vol. 14, pp. 409-421, 2011.
- [5] Ravizza, S., Chen, J., Atkin, J. A. D., Burke, E. K., and Stewart, P., “The trade-off between taxi time and fuel consumption in airport ground movement,” *Public Transport*, Vol. 5, pp. 25-40, 2013.
- [6] Mori, R., “Optimal Spot-out Time – Taxi-out Time Saving and Corresponding Delay,” *Proceedings of 4th CEAS Air & Space Conference*, 2013.
- [7] Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.

8 Contact Author Email Address

r-mori@enri.go.jp

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS 2014 proceedings or as individual off-prints from the proceedings.